Cheng Zhang

Curriculum vitae

czhang03@bu.edu
cs-people.bu.edu/czhang03/
czhang03
Full Curriculum vitae

Education

2018 — 2024 **Computer Science, Doctor of Philosophy**, Boston University, Boston, MA

Primary Interest: Kleene Algebra and its application in program verification

I am broadly interested in application of mathematics in computer science, especially in programming languages. I have worked on Kleene algebra, automata thoery, semantics, type systems, and their application in program verification.

2014 — 2018 **Mathematics, Bachelor of Art**, *with department honor, magna cum laude*, Wheaton College, Norton, MA

Honor Thesis: King in Generalized Tournaments.

Minors: Computer Science, Economics.

2016 — 2017 **Economics, Study Aboard**, London School Of Economics, London, United Kingdom

Preprints And Drafts

2024 **Cheng Zhang, Hang Ji, Ines Santacruz, Marco Gaboardi**, *A Symbolic Decision Procedure For GKAT, and its Complexity*, Draft

2024 **Cheng Zhang, Tobias Kappé, David E. Narváez, Nico Naus**, *CF-GKAT: Control Flow Verification in Nearly Linear Time*, Draft

2024 **Arthur Azevedo de Amorim, Cheng Zhang, Marco Gaboardi**, *Kleene algebra with commutativity conditions is undecidable*, Preprint

Publications

2024 **Cheng Zhang, Arthur Azevedo de Amorim, Marco Gaboardi**, *Domain Reasoning In TopKAT*, International Colloquium on Automata, Languages and Programming (ICALP)

2023 **Mark Lemay, Qiancheng Fu, William Blair, Cheng Zhang, Hongwei Xi**, *A Dependently Typed Language with Dynamic Equality*, The workshop on Type-Driven Development (TyDe)

2022 **Cheng Zhang, Arthur Azevedo de Amorim, Marco Gaboardi**, *On Incorrectness Logic and Kleene Algebra With Top and Tests*, Principle Of Programming Language (POPL)

2020 **Mark Lemay, Cheng Zhang, William Blair**, *Developing a Dependently Typed Language with Runtime Proof Search (Extended Abstract)*, Workshop on Type-Driven Development (TyDe)

2018 **Cheng Zhang**, *King in Generalized Tournaments*, Wheaton College Honor Thesis

2018 **Cheng Zhang, Weiqi Feng, Emma Steffens, Alvaro de Landaluce, Scott Kleinman, Mark D. LeBlanc**, *Lexos 2017: Building Reliable Software in Python*, Journal of Computing Sciences in Colleges

Research Talks

2023 **Cheng Zhang**, *GKAT with Indicator Variables, Fast Decompilation Verification*, BU Principles of Programming and Verification (POPV) Seminar

2023 **Cheng Zhang**, *A Practical Tutorial to KAT and its Extensions*, Systems Software Research Group at Virginia Tech

2022 **Cheng Zhang**, *Kleene Algebra and Its Applications in Verification*, Boston Computation Club

2022 **Cheng Zhang**, *On Incorrectness Logic Kleene Algebra With Test*, Cornell Programming Language Discussion Group (PLDG)

2022 **Cheng Zhang**, *On Incorrectness Logic Kleene Algebra With Test*, Principle Of Programming Languages (POPL)

2018 **Cheng Zhang, Mark D. LeBlanc**, *Lexos 2017: Building Reliable Software in Python*, Journal of Computing Sciences in Colleges

2018 **Cheng Zhang**, *Kings in Quasi-transitive Oriented Graph*, Wheaton Summit For Woman In STEM

Research Projects

2024 **A Symbolic Decision Procedure For Guarded Kleene Algebra With Tests**

Guarded Kleene Algebra With Tests is an efficient framework for program verifications. In this work, We marry GKAT with symbolic automaton, further improving its efficiency. Unlike previous works with regard to KAT, our algorithm was able to offload the solving of complex boolean logic to a trusted solver like z3. This work also gives the exact complexity bound of GKAT.

2024 **Guarded Kleene Algebra With Tests And Control Flow Verification**

Correctness of compilation and decompilation remains a challenging step in verifying software correctness, one of the most challenging steps remains to be control flow restructuring. Many works in compiler and decompiler do not verify the correctness of their algorithm, stumped by the complexity of the verification process. In this work we designed an algorithm to formally verify control-flow restructuring. This algorithm is not only sound and complete with regard to trace equivalences, but also able to handle a wide range of control-flow structures like goto, break, return, and indicator variables. Furthermore, we provide an experimental implementation, allowing developers to interface with our algorithm, and we have verified that this algorithm was able to verify program with millions of command in seconds, running on a laptop.

2022 — 2024 **Kleene Algebra With Commutative Hypothesis**

Researchers has long known the importance of commutative hypothesis in Kleene Algebra. However, with recent work, commutative hypothesis was put on the center stage as the foundation of relational verifications. Unfortunately, the theory of KA with commutativity hypothesis is not as well-developed as its applications. Specifically, the decidability of Kleene algebra with commutativity hypothesis has remained open for almost three decades. In this work, we took on this long-standing open problem, and proven that the Kleene Algebra with commutativity hypothesis is undecidable. In fact, we were able to obtain our undecidability result with only the left unfolding rule, and omitted induction rules. This means a large class of theories sitting between idempotent semiring with left unfolding, and the regular languages are all undecidable with commutativity hypothesis.

2021 — 2022 **Probabilistic Incorrectness Logic and Kleene Algebra**

Examine the mathematical foundation of probabilistic Incorrectness logic. This work may give rise to a method reason about correctness and incorrectness in probabilistic programs.

2020 — 2022 **Algebraic Formulation Of Incorrectness Logic**

Provide an algebraic formulation of Incorrectness Logic in TopKAT. Our work leads to simpler proofs for program incorrectness, and demonstrates ways to automatically certify bugs in programs. We showed that TopKAT is a minimal framework to model incorrectness, as it is impossible to encode incorrectness logic in KAT. After that, we proved many meta-theoretical property of TopKAT, including incompleteness with relational model, completeness of general relational model and language model, complexity of deciding equality, and expressivity of general relational model.

2017 — 2018 **Mathematics Honor Thesis**, *Wheaton College*, Norton, MA

Studied kings in generalizations of tournament, with a special focus on quasi-transitive oriented graphs. I have shown that all the quasi-transitive oriented graphs can be condensed into a tournament via tie component condensation, and tie component condensation of quasi-transitive oriented graphs is the most efficient condensation to tournament.

2015 — 2018 **Software Lead**, *Lexomics Research Group, Wheaton College*, Norton, MA

Led a group of undergraduate engineers through a major factorization of the natrual language processing (NLP) software Lexos. In the process, I have designed a new architecture for side-effect management, transitioned the code base to a scalable functional-first paradigm, implemented industry-standard software development workflows, and provided detailed documentations and guides for the entire system.

Honors And Fellowships

2022 Meta PhD Research Fellowship Finalist in Programminag Languages

2018 — Now Phi Beta Kappa Honor Society Member.

2018 Boston University Dean's Fellowship.

2018 Phi Beta Kappa Graduate Scholarship.

2018 Madeleine F. Clark Wallace Mathematics Prize.

2018 Fred Kollett Prize in Mathematics & Computer Science.

2017 Weaton College Faculty-Student Research Awards.

2016 Wheaton Fellows.

2014 — 2018 Wheaton College International Scholarship.

2014 — 2018 Wheaton College Dean's Lists.

Professional Experiences And Service

2023 **Artifact Reviewer For ESOP (European Symposium on Programming) 2023**

2019 — Now **Graduate Researcher**, *Boston University*, Boston, MA

Study various extensions of Kleene Algebra, and their use in program analysis of imperative/functional programs, probabilistic programs, distributed systems, networks, etc. My researches provide easier, even automated, proofs for program analysis.

2019 — 2021 **Teaching Fellow**, *Boston University*, Boston, MA

Taught Principle of Programming Language, Introduction to Computer Science, Algebra Algorithm, Geometric Algorithm, etc.

2021 — Now **Organizer**, *Principle of Programming and Verification Seminar*, Boston University, Boston, MA

Invite seminar speakers and coordinate time of the talks; maintain seminar webpage, mailing list, and calendar; distribute details of the seminar to participants every week; and host speakers during the seminars.

2020 — Now **Organizer**, *Programming Language Reading Group*, Boston University, Boston, MA

Identify and distribute weekly reading materials; host and schedule the weekly discussions.

2019 **Grader**, *Boston University CS 511 Formal Method*, Boston, MA

Provided solutions to homework problems, graded the homework, organize useful statistics for the professor, and provided hints and answered questions piazza when necessary.

2017 — 2018 **Grader**, *Wheaton College MATH 241 Theory of Probability*, Norton, MA

Graded homework, gave feedback to students on each individual homework, and provide informative statistics to the professor on the homework.

Selected Coursework

2018 — Now **Computer Science, Doctor of Philosophy**, Boston University

○ Computer Network
○ Formal Method
○ Overview On Type Systems
○ Cryptography
○ Compilers
○ Complexity Thoery

2014 — 2018 **Mathematics, Bachelor of Art**, Wheaton College

○ Complex Analysis
○ Graph Thoery
○ Mathematical Statistics
○ Real Analysis
○ Theory Of Computation
○ Advanced Cryptography

2016 — 2017 **Economics, Study Aboard**, London School Of Economics

○ Game Thoery
○ Abstract Algebra I
○ Econromatrics
○ Abstract Algebra II